



## UMA MATEHEURÍSTICA PARA O PROBLEMA DE PLANEJAMENTO DE CIRCUITOS VLSI

**LETICIA LEITE PAVANELLO** – leticia.pavanello@unesp.br  
UNIVERSIDADE ESTADUAL PAULISTA – UNESP – BAURU

**CARLOS DIEGO RODRIGUES** - diego@lia.ufc.br  
UNIVERSIDADE FEDERAL DO CEARÁ – UFC – FORTALEZA

**ADRIANA CRISTINA CHERRI** – adriana.cherri@unesp.br  
UNIVERSIDADE ESTADUAL PAULISTA – UNESP – BAURU

**Área:** 3. PESQUISA OPERACIONAL

**Sub-Área:** 3.1 – MODELAGEM, SIMULAÇÃO E OTIMIZAÇÃO

**Resumo:** O PROBLEMA DE PLANEJAMENTO DE CIRCUITOS VLSI ABORDADO CONSISTE EM DETERMINAR A LOCALIZAÇÃO FÍSICA DE DISPOSITIVOS ELETRÔNICOS DENTRO DOS LIMITES DE UM CHIP, SEM QUE HAJA SOBREPOSIÇÕES. COM O OBJETIVO DE MINIMIZAR O COMPRIMENTO DO FIO USADO EM SUAS INTERCONEXÕES, É PRECISO IDENTIFICAR QUAIS ITENS DEVEM SER COLOCADOS PRÓXIMOS UNS AOS OUTROS E ALOCÁ-LOS DE FORMA A ATENDER METAS, MUITAS VEZES CONFLITANTES, DE ESPAÇO DISPONÍVEL E DESEMPENHO. DEVIDO SUA NATUREZA COMBINATÓRIA E FORTE COMPONENTE GEOMÉTRICA, É UM PROBLEMA DIFÍCIL DE SER RESOLVIDO. QUANDO CONSIDERADAS GRANDES INSTÂNCIAS, OBTENÇÃO DE SOLUÇÕES EXATAS TORNA-SE AINDA MAIS CUSTOSO E INVIÁVEL. DESTA FORMA, MÉTODOS HEURÍSTICOS SÃO COMUMENTE CONSIDERADOS PARA RESOLVER ESTE TIPO DE PROBLEMA, OBTENDO BOAS APROXIMAÇÕES DA SOLUÇÃO ÓTIMA COM UM BAIXO ESFORÇO COMPUTACIONAL. NESTE TRABALHO, UMA ABORDAGEM “METEHEURÍSTICA” DE JANELA DESLIZANTE É APRESENTADA PARA RESOLVER O PROBLEMA DE MANEIRA EFICIENTE, COMBINANDO PROGRAMAÇÃO MATEMÁTICA, UMA ABORDAGEM HEURÍSTICA E ESTRATÉGIAS DE ORDENAÇÃO DE ITENS COM DIMENSÕES FLEXÍVEIS. PARA AVALIAR O DESEMPENHO DO PROCEDIMENTO, TESTES COMPUTACIONAIS FORAM REALIZADOS COM CINCO INSTÂNCIAS DE REFERÊNCIA DO CONJUNTO MCNC NA LINGUAGEM DE PROGRAMAÇÃO C++ COM O SOLVER CPLEX. OS RESULTADOS DEMONSTRAM GRANDE INFLUÊNCIA DA ESTRATÉGIA DE ORDENAÇÃO NA QUALIDADE DA SOLUÇÃO OBTIDA E POTENCIAL DO MÉTODO.

**Palavras-chaves:** PROBLEMAS DE EMPACOTAMENTO; RETÂNGULOS FLEXÍVEIS; DESIGN VLSI; HEURISTICA; JANELA DESLIZANTE.

**Abstract:** *THE VLSI CIRCUIT FLOORPLANNING ADDRESSED CONSISTS OF DEFINE THE PHYSICAL PLACEMENT OF ELECTRONIC DEVICES WITHIN THE LIMITS OF A CHIP, WITHOUT OVERLAPPING. IN ORDER TO MINIMIZE THE WIRE LENGTH USED IN ITS INTERCONNECTIONS, IT'S NECESSARY TO IDENTIFY WHICH ITEMS SHOULD BE PLACED CLOSE TO EACH OTHER AND ALLOCATE THEN TO MEET GOAL, OFTEN CONFLICTING, SUCH AS AVAILABLE SPACE AND PERFORMANCE. DUE TO ITS COMBINATIONAL NATURE AND STRONG GEOMETRIC COMPONENT, IT'S A DIFFICULT PROBLEM TO BE SOLVED. WHEN LARGE INSTANCES, OBTAINING EXACT SOLUTIONS BECOME EVEN MORE IMPRACTICABLE AND COSTLY. THUS HEURISTIC METHODS ARE COMMONLY CONSIDERED TO SOLVE THIS TYPE OF PROBLEM, OBTAINING GOOD APPROACHES TO THE OPTIMUM SOLUTION WITH LOW COMPUTATIONAL EFFORT. IN THIS WORK, A "MATHEHEURISTIC" SLIDING WINDOW APPROACH IS PRESENTED TO EFFICIENTLY SOLVE THE PROBLEM, COMBINING MATHEMATICAL PROGRAMMING, A HEURISTIC APPROACH AND RANKING STRATEGIES FOR FLEXIBLE ITEMS. TO EVALUATE THE PERFORMANCE OF THE PROCEDURE, COMPUTATIONAL TESTS WERE PERFORMED WITH FIVE REFERENCE INSTANCES OF THE MCNC SET IN THE C++ PROGRAMMING LANGUAGE WITH THE CPLEX SOLVER. THE RESULTS DEMONSTRATE GREAT INFLUENCE OF THE ORDERING STRATEGY ON THE QUALITY OF THE SOLUTIONS OBTAINED AND THE POTENTIAL OF THE METHOD.*

**Keywords:** *PACKING PROBLEMS; SOFT RECTANGLES; VLSI DESIGN; HEURISTIC; SLIDING WINDOW.*

## 1. INTRODUÇÃO

Os problemas de empacotamento são estudados há décadas como uma ferramenta de apoio na tomada de decisão em diversos setores. Com uma forte característica geométrica, os problemas de empacotamento bidimensionais consistem em empacotar retângulos menores (itens) com dimensões predefinidas em regiões retangulares maiores utilizando os recursos disponíveis da melhor forma possível.

Uma variação para o problema de empacotamento bidimensional ocorre quando os retângulos a serem empacotados possuem dimensões flexíveis. Nesse caso, a área de cada item pode variar em um determinado intervalo, assim como a proporção entre suas respectivas alturas e larguras. Os problemas de empacotamento de retângulos flexíveis são problemas de otimização combinatória difíceis de resolver, pois neste caso é necessário selecionar como os retângulos de um conjunto pré-definido serão empacotados na região e, ainda, decidir quais as suas dimensões, respeitando as suas limitações geométricas e a não-sobreposição dos itens.

Além de sua importância teórica, este problema surge em várias situações práticas e está presente na determinação do layout de circuitos de integração de grande escala. O VLSI (*Very Large Scale Integration*) é uma tecnologia de microeletrônica que integra uma grande quantidade de dispositivos numa pastilha de silício, conhecida como chip. O processo de determinar a localização física dos dispositivos e interconectá-los dentro dos limites do chip utilizando os recursos disponíveis do melhor modo possível, é um problema de empacotamento com retângulos, que podem ter suas dimensões flexíveis.

Com o objetivo de minimizar o comprimento do fio usado em suas interconexões, é preciso identificar quais itens devem ser colocados próximos uns aos outros e alocá-los de forma a atender metas, muitas vezes conflitantes, de comprimento de fio utilizado, espaço disponível no chip e desempenho necessário. Com blocos posicionados de acordo com a descrição detalhada de conectividade do circuito, obtém-se distâncias de interconexão mais curtas, que utilizam menos recursos de roteamento, os caminhos de sinal ponta a ponta são mais rápidos e tempos de rota mais consistentes são possíveis.

Neste trabalho, uma abordagem mateheurística (*matheuristic*) de janela deslizante é proposta para resolver de maneira eficiente o problema de planejamento de circuitos VLSI. Combinando programação matemática e um método heurístico, este procedimento iterativo conta com uma estratégia de inicialização que é estabelecida utilizando um critério de ordenação para os dispositivos que se deseja alocar e um sub-modelo é criado para resolver a

alocação de parte dos retângulos do problema original, diminuindo significativamente a complexidade computacional do modelo. A influência de diferentes critérios na ordenação inicial sobre o resultado final será analisada.

A Seção 2 deste trabalho é composta por uma breve revisão de literatura. O modelo matemático é abordado na seção 3 e o procedimento proposto é detalhado na Seção 4. Na Seção 5, resultados de testes computacionais são apresentados, e a Seção 6 destina-se às conclusões e trabalhos futuros.

## 2. REVISÃO DE LITERATURA

Abordagens exatas para o problema de empacotamento bidimensional com retângulos rígidos foram propostas por Fekete e Schepers (2000), Martello e Monaci (2015), entre outras. Entretanto, devido a resolução custosa do problema, muitas abordagens de solução são heurísticas. A heurística mais antiga e famosa é a *bottom-left* (BL), proposta por Baker et al. (1980). Murata e Kuh (1998) propuseram um método para a alocação de um conjunto de módulos mistos flexíveis, baseado na representação por pares de sequência (*sequence pair*).

Young et al. (2001) utilizaram relaxação lagrangeana para resolver o problema de alocação de módulos retangulares flexíveis. O método desenvolvido teve por objetivo minimizar a área total de empacotamento dos módulos, e foi implementado em um *simulated annealing framework* utilizando a representação por pares de sequência.

Com uma abordagem semelhante, Chi e Chi (2002) propuseram um algoritmo *floorplanning* com módulos flexíveis para otimizar a área utilizada no design de um circuito VLSI. Ibaraki e Nakamura (2006) abordaram o problema de empacotamento de retângulos, cujas formas poderiam ser ajustáveis, dentro de determinadas restrições de perímetro e área. Os autores utilizaram pares de sequência para especificar posições relativas entre cada par de retângulos. Para determinar o tamanho exato e a localização de todos os retângulos, uma formulação matemática foi proposta e para encontrar bons pares de sequências, técnicas de busca local foram utilizadas.

Chen e Chang (2009) propuseram uma abordagem tratando as posições relativas dos módulos retangulares flexíveis como variáveis binárias do problema no *floorplanning*. Chan et al. (2013) adotaram uma abordagem de dois estágios para o *floorplanning* de um conjunto de módulos de tamanhos variados. O primeiro estágio consiste na distribuição dos módulos em um contorno fixo otimizando o comprimento de fio utilizado. No segundo estágio, é realizada uma abordagem baseada em partição para obter uma solução viável sem prejudicar o

bom resultado atingido no estágio inicial.

Laskar et al. (2015) apresentaram um estudo e comparação dos diferentes algoritmos de otimização e as representações envolvidas no problema de *layout* do VLSI. Dentre os algoritmos, a maior parte do trabalho foi realizada para otimização de minimização de área. Singh et al. (2016) reuniram e revisaram heurísticas e algoritmos meta-heurísticos de trabalhos que consideram o problema de otimização no *floorplanning* de VLSI. Comparações entre os procedimentos reunidos foram incluídas neste estudo e mesmo a representação mais eficaz apresentada indicou limitações de flexibilidade.

Wu et al. (2016) propuseram um procedimento heurístico para resolver o problema de minimização da área de empacotamento de retângulos quando o *layout* final deveria conter um retângulo central, com flexibilidade nas dimensões, as quais poderiam ser alteradas dentro de limites razoáveis. Estratégias de preenchimento do espaço interno foram propostas para melhorar a taxa de preenchimento do *layout* final.

Jenifer et al. (2016) propuseram uma abordagem para a determinação do design físico do VLSI. Com o objetivo de obter a área mínima do *floorplanning* remodelando os blocos presentes, o trabalho proposto redefine o problema com uma meta-heurística baseada em simulated annealing.

Apesar dos diversos trabalhos abordando o problema de empacotamento de retângulos, métodos de solução mais eficientes podem ser explorados, garantindo que muitos estudos ainda sejam desenvolvidos e possibilitando o uso de instâncias mais próximas do ponto de vista prático, principalmente no contexto do posicionamento de módulos flexíveis em circuitos VLSI.

### 3. MODELO MATEMÁTICO

O problema de posicionamento de módulos flexíveis em circuitos VLSI considera como variáveis de decisão as posições ideais para um determinado conjunto de módulos alocados em uma área de chip, de modo que não haja sobreposição, além de decidir quais as dimensões mais adequadas para cada retângulo flexível. Tem como objetivo minimizar as distâncias do comprimento de fio de meio-perímetro ponderado (HPWL) entre subconjuntos especificados dos módulos, isto é, reduzir a medida da quantidade de metal necessária para interconectar os circuitos de forma a garantir a funcionalidade do chip. Os parâmetros e variáveis do modelo matemático, adaptação feita do modelo de Chen e Chang (2009) são definidos por:

### Parâmetros:

- $n$ : número de blocos flexíveis.
- $P$ : número de redes.
- $A$ : conjunto de componentes  $a$ ,  $a \in \{1, \dots, n\}$  com dimensões máximas, mínimas e proporções máximas e mínimas.
- $R$ : conjunto de redes  $r$ ,  $r \in \{1, \dots, p\} \subseteq \mathcal{P}(A)$ , subconjuntos de  $A$ .

### Variáveis:

- $x_a, y_a$ : posição do bloco  $a$ .
- $w_a, h_a$ : dimensões do bloco  $a$ .
- $W_r, H_r$ : dimensões da rede  $r$ .
- $p_{ab}$ : relações horizontais de posição para todo  $a, b \in A$  sendo  $a \neq b$ .
- $q_{ab}$ : relações verticais de posição para todo  $a, b \in A$  sendo  $a \neq b$ .

Variáveis binárias que indicam a posição do bloco  $a$  em relação ao bloco  $b$

### Modelo:

$$\text{minimizar } f(x) = \sum_{r \in R} W_r + H_r$$

$$\text{Sujeito a:} \tag{1}$$

$$w_a^{\min} \leq w_a \leq w_a^{\max}, \forall a \in A \tag{2}$$

$$h_a^{\min} \leq h_a \leq h_a^{\max}, \forall a \in A \tag{3}$$

$$s_a^{\min} \cdot h_a \leq w_a \leq s_a^{\max} \cdot h_a, \forall a \in A \tag{4}$$

$$x_a \geq x_b + w_b - (1 - p_{ab}) \cdot W^{\max}, \text{ se } p_{ab} = 1, \forall a \neq b \in A \tag{5}$$

$$y_a \geq y_b + h_b - (1 - q_{ab}) \cdot H^{\max}, \text{ se } q_{ab} = 1, \forall a \neq b \in A \tag{6}$$

$$p_{ab} + q_{ab} + p_{ba} + q_{ba}, \forall a < b \in A \tag{7}$$

$$W_r \geq \left(x_b + \frac{w_b}{2}\right) - \left(x_a + \frac{w_a}{2}\right), \forall r \in R \text{ e } \forall a, b \in r \tag{8}$$

$$H_r \geq \left(y_b + \frac{h_b}{2}\right) - \left(y_a + \frac{h_a}{2}\right), \forall r \in R \text{ e } \forall a, b \in r \tag{9}$$



No modelo (1)-(8), a função objetivo (1) minimiza o comprimento de fio utilizado nas interconexões dos blocos adotando a simplificação de distância entre dois módulos, que é sempre calculada entre os centros geométricos dos módulos neste caso, ao invés dos pinos, conforme as instâncias originais. As restrições (2), (3) e (4) garantem que as dimensões dos blocos flexíveis estejam dentro do intervalo permitido, respeitando a proporção  $ha$ . As restrições (5), (6) e (7) se referem às relações de posição entre os itens e garantem que apenas uma das relações seja igual a 1 para cada par de blocos, assim não é possível que um bloco esteja, simultaneamente, à direita e à esquerda de outro bloco. Por fim, as restrições (8) e (9) são sobre as dimensões das redes.

#### 4. ABORDAGEM MATEHEURÍSTICA

O problema de posicionamento de módulos flexíveis em circuitos VLSI pode ser facilmente representado através de modelos matemáticos, porém é difícil de ser solucionado devido ao grande número de variáveis envolvidas. Obter soluções exatas implica em alta complexidade computacional, principalmente quando consideradas grandes instâncias. Assim, métodos heurísticos são utilizados para obter boas aproximações de forma menos custosa, se comparadas com a utilização de métodos exatos. Nesta seção, uma abordagem mateheurística (*matheuristic*) de janela deslizante é proposta para resolver de maneira eficiente o problema de planejamento de circuitos VLSI.

Combinando programação matemática e um método heurístico, o procedimento iterativo proposto conta com uma estratégia de inicialização que é estabelecida utilizando um critério de ordenação para os dispositivos que se deseja alocar. Neste passo, ordenam-se os retângulos por algum critério conveniente, possivelmente aleatório. Um sub-modelo é criado para resolver a alocação de parte dos retângulos do problema original, diminuindo significativamente a complexidade computacional do modelo. Apenas os  $j$  primeiros itens da lista ordenada são considerados no sub-modelo, sendo  $j$  um parâmetro do modelo, conhecido como tamanho da janela.

O sub-modelo é resolvido e, a partir da sub-solução obtida, fixam-se as posições relativas referentes ao item  $i$ , sendo  $i$  o índice da iteração presente, e atualiza-se o sub-modelo acrescentando o próximo item não-alocado da lista ordenada. O sub-modelo é resolvido novamente. Estes passos são realizados de modo iterativo até que a quantidade de itens do sub-modelo seja igual à quantidade de itens do modelo inicial. Adota-se como solução do modelo a solução obtida para o sub-modelo na última iteração realizada.

## ALGORITMO

---

### Passo 1. Inicialização

Ordene os blocos que se deseja alocar utilizando um critério de ordenação.

### Passo 2. Sub-modelo

Sendo  $j$  uma constante pré-estabelecida e  $i$  o índice da iteração presente. Até que a quantidade de itens  $(j + i)$  do sub-modelo seja igual a quantidade de itens do modelo original, faça:

- Um sub-modelo é criado para resolver a alocação dos itens considerando apenas uma quantidade limitada de  $j$  blocos variáveis;
- Resolva o sub-modelo com  $(j + i)$  itens;
- Fixam-se as posições relativas referentes ao item  $i$ ;
- Atualize o sub-modelo acrescentando o item de índice  $(j + i)$  da lista ordenada.

### Passo 3. Solução do modelo

Adota-se como solução do modelo a solução obtida para o sub-modelo na última iteração realizada.

---

## 5. TESTES COMPUTACIONAIS

Para avaliar o desempenho do procedimento proposto, testes computacionais foram realizados na linguagem de programação C++ com o solver CPLEX e executados em um computador com processador Intel Core i7-3517U com 8GB de memória RAM. Foram utilizadas cinco instâncias do conjunto de referência arquivado no *Microelectronics Center of North Carolina* (MCNC) com origem no projeto de circuitos VLSI, no qual todos os itens têm forma retangular flexível e o número de itens a serem posicionados não ultrapassam os 50.

As instâncias foram testadas para três diferentes valores do parâmetro  $j$ , que corresponde ao tamanho da janela deslizante no modelo e indica a quantidade de variáveis livres a serem consideradas por iteração, variando nos valores  $J = [1, 3, 5]$ . Ainda, cinco critérios de ordenação foram considerados na inicialização dos itens: (a) área de modo não-decrescente, (b) área de modo não-crescente, (c) conectividade não-decrescente, (d) conectividade não-crescente e, por fim, (e) ordenação aleatória. Foi estipulado o limite de tempo de 300 segundos para a obtenção de cada solução no sub-modelo.

As Tabelas 1 a 5 apresentam os valores de comprimento de fio e tempo computacional obtidos pela mateheurística para cada instância, combinando diferentes critérios de ordenação inicial e tamanhos de janela. Destaca-se o melhor resultado obtido para a instância.



TABELA 1 – Comprimento de fio e tempo computacional por ordenação para a instância AMI33

	j = 1		j = 3		j = 5	
<b>AMI33</b>	<b>Solução</b>	<b>Tempo</b>	<b>Solução</b>	<b>Tempo</b>	<b>Solução</b>	<b>Tempo</b>
Menor área (a)	26862,5	97,797	24157	6496,74	23786	22632,5
Maior área (b)	28546	30,969	24437	2936,82	25928	16784,5
Menor conectividade (c)	16492	121,361	<b>15393</b>	11503,3	15515,5	20845
Maior conectividade (d)	23590	148,549	21189	8349,35	22606,5	14821,7
Aleatório (e)	19411	390,116	26992	4106,48	30156	17644,9

TABELA 2 – Comprimento de fio e tempo computacional por ordenação para a instância AMI49

	j = 1		j = 3		j = 5	
<b>AMI49</b>	<b>Solução</b>	<b>Tempo</b>	<b>Solução</b>	<b>Tempo</b>	<b>Solução</b>	<b>Tempo</b>
Menor área (a)	356814	1055,4	436484	11056,4	464532	10982,6
Maior área (b)	403112	191,598	421536	580,204	402972	3184,63
Menor conectividade (c)	344452	246,361	308557	3108,19	<b>306202</b>	12574,9
Maior conectividade (d)	362426	409,797	437191	4189,91	355374	7201,85
Aleatório (e)	386646	699,936	374428	1818,78	339504	8897,01

TABELA 3 – Comprimento de fio e tempo computacional por ordenação para a instância APTE

	j = 1		j = 3		j = 5	
<b>APTE</b>	<b>Solução</b>	<b>Tempo</b>	<b>Solução</b>	<b>Tempo</b>	<b>Solução</b>	<b>Tempo</b>
Menor área (a)	130074	4,893	124739	47,281	116978	137,865
Maior área (b)	197494	4,627	126250	32,127	126250	435,431
Menor conectividade (c)	122450	4,559	108274	28,729	<b>104486</b>	641,047
Maior conectividade (d)	165630	6,541	161028	26,559	159086	743,925
Aleatório (e)	91087	5,522	101235	6,693	102386	661,704

TABELA 4 – Comprimento de fio e tempo computacional por ordenação para a instância HP

	j = 1		j = 3		j = 5	
<b>HP</b>	<b>Solução</b>	<b>Tempo</b>	<b>Solução</b>	<b>Tempo</b>	<b>Solução</b>	<b>Tempo</b>
Menor área (a)	61299	4,333	50925	56,88	44643	999,131
Maior área (b)	47090	3,638	34979,5	37,969	34395	914,076
Menor conectividade (c)	49349	4,999	41198,5	45,393	38448,5	232,798
Maior conectividade (d)	41378	3,611	32529	98,133	<b>31203</b>	1013,09
Aleatório (e)	40819	3,617	39786,5	35,722	43291	764,755

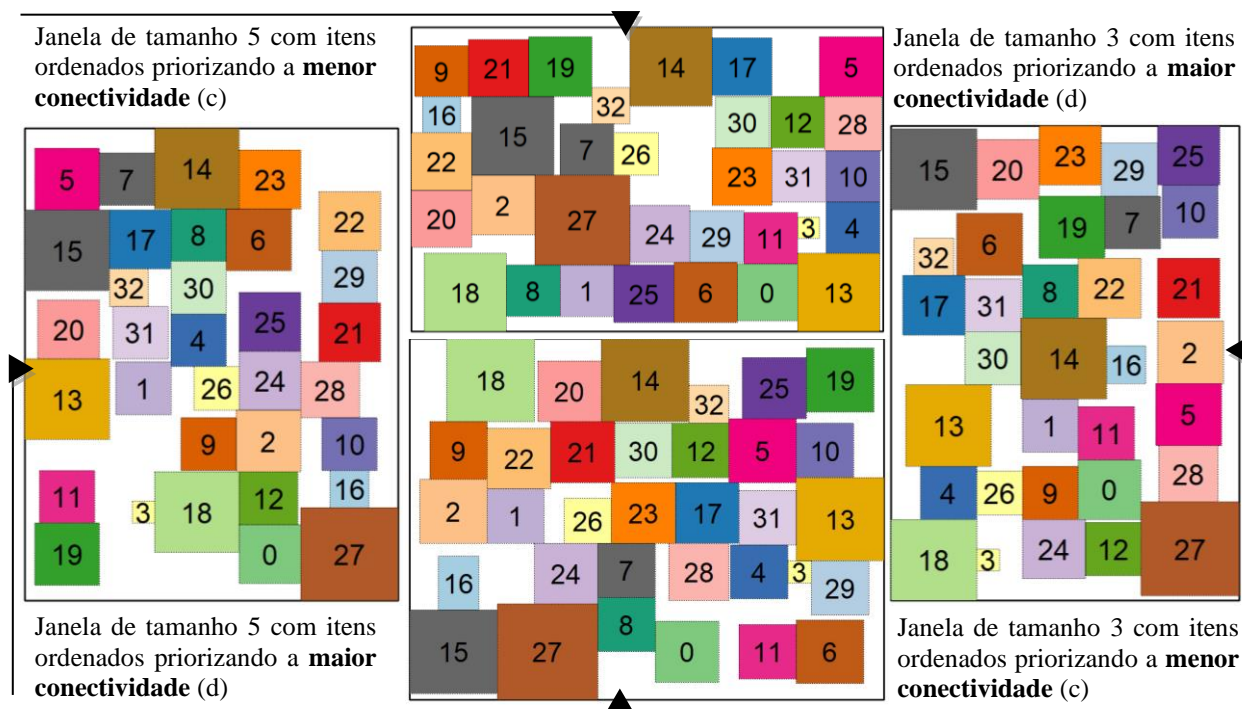
TABELA 5– Comprimento de fio e tempo computacional por ordenação para a instância XEROX

	j = 1		j = 3		j = 5	
<b>XEROX</b>	<b>Solução</b>	<b>Tempo</b>	<b>Solução</b>	<b>Tempo</b>	<b>Solução</b>	<b>Tempo</b>
Menor área (a)	413807	5,689	410370	23,386	381810	628,338
Maior área (b)	413008	4,837	321813	8,191	323108	687,003
Menor conectividade (c)	358890	5,461	<b>315004</b>	16,304	327607	274,853
Maior conectividade (d)	528906	5,804	448472	9,282	390386	170,678
Aleatório (e)	452046	7,092	269290	22,173	314802	528,961

Pela Tabela 4 observa-se que a melhor solução encontrada para a instância HP foi obtida ao adotar o critério de ordenação por conectividade de modo não-crescente, considerando uma janela deslizando de tamanho 5. Para a instância AMI49, o melhor resultado obtido considerou como o critério de inicialização a conectividade de itens modo não-decrescente, característica que se manteve para as demais instâncias: APTE, XEROX e AMI33.

Ao definir um critério de ordenação inicial é possível estabelecer quais itens serão priorizados na alocação feita e, com isso, melhorar o processo iterativo realizado. De acordo com os dados supracitados nas Tabelas 1 a 5, para todas as instâncias testadas, os melhores resultados para os testes foram obtidos ao utilizar, como estratégia, ordenações baseadas nas características da descrição detalhada da conectividade do VLSI.

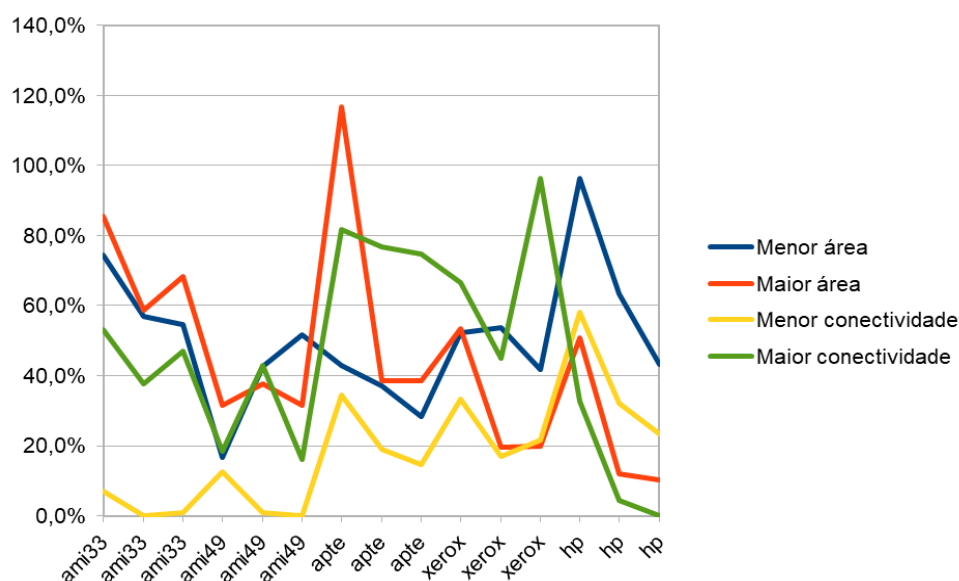
FIGURA 1 – Resultado visual de alocações obtidas para AMI33 com diferentes tamanhos de janela e ordenações



As soluções obtidas indicam grande influência do critério de ordenação no resultado final do problema. Com tendência a obter o melhor resultado, ordenar os itens pela conectividade de modo não-decrescente garante que o processo de alocação seja iniciado fixando primeiramente os itens que possuem menor conectividade e flexibilizando o posicionamento dos itens que aparecem em um número maior de conexões. Na Figura 1 é possível visualizar a solução geométrica de algumas alocações obtidas para a instância AMI33 com diferentes tamanhos de janela e ordenações por conectividade.

O Gráfico 1 ilustra a comparação de soluções entre as instâncias para cada critério de ordenação. No eixo vertical é expresso o percentual que indica a distância da solução em relação a melhor solução obtida para aquela instância. No eixo horizontal, cada instância é indicada pelo seu nome seguindo, respectivamente, o tamanho de janela  $j$  igual a 1, 3 e 5. O critério que prioriza itens com menor conectividade fica em destaque, com valores inferiores ao demais critérios nas instâncias AMI33, AMI49, APTE e XEROX.

GRAFICO 1 – GAP de soluções entre instâncias por critério de ordenação



Ainda, vale observar a relação entre o tempo computacional e o tamanho de janela adotado. Como proposto na abordagem, ao criar sub-modelos com uma quantidade limitada de variáveis livres, a complexidade computacional é reduzida e o tempo de execução diminui consideravelmente. Tamanhos de janelas maiores tendem a gerar resultados melhores, porém com tempos de processamento muito mais elevados, nem sempre viáveis. Por exemplo, na Tabela 2, para a instância AMI49 com ordenação por conectividade não-decrescente, a

solução para janela de tamanho 5 é cerca de 12% melhor do que a solução para janela 3, porém o tempo computacional sofre um aumento de 510%.

## 6. CONCLUSÃO

Neste artigo, uma abordagem mateheurística de janela deslizante foi proposta para a obtenção de soluções para o problema de planejamento de circuitos VLSI. Através de uma estratégia inicial de ordenação dos módulos que se deseja alocar e da criação de sub-modelos com variáveis livres limitadas, foi possível diminuir significativamente a complexidade computacional do modelo e encontrar soluções satisfatórias para o problema.

Os resultados demonstram que bons critérios de inicialização têm grande influência sobre a qualidade da solução e estão intimamente ligados às características de conectividade do circuito. Explorar melhores conjuntos de ordenação inicial pode economizar tempo e processamento durante a execução da abordagem, de forma simples e pouco custosa. Testes computacionais foram realizados com conjuntos de referência na linguagem de programação C++ com o solver CPLEX e indicam a eficiência do método apresentado.

Para trabalhos futuros é sugerida a exploração de novos critérios de inicialização, bem como a implementação de uma estratégia de finalização, com o objetivo de lapidar a solução encontrada com possíveis ajustes nos itens livres permitidos pela metaheurística. Além disso, testes com diferentes tamanhos de janela podem ser explorados, aumentando o limite estipulado para tempo máximo de execução de cada sub-modelo.

## AGRADECIMENTOS

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela credibilidade e apoio financeiro.

## REFERÊNCIAS

- BAKER, B. S.; COFFMAN, JR, E. G.; RIVEST, R. L. (1980). Orthogonal packings in two dimensions. *SIAM Journal on computing*, 9(4), 846–855.
- CHAN, K.; HSU, C.; LIN, J. (2013). A flexible fixed-outline floorplanning methodology for fixed-size modules. *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 18.
- CHEN, T.-C.; CHANG, Y.-W. (2009). *Floorplanning in Electronic Design Automation*, pages

575–634. Elsevier.

CHI, J.; CHI, M. (2002). An effective soft module floorplanning algorithm based on sequence pair. IEEE International ASIC/SOC Conference, 15.

FEKETE, S. P.; SCHEPERS, J. (2000). On more-dimensional packing iii: Exact algorithms.

HUANG, W.; CHEN, D.; XU, R. (2007). A new heuristic algorithm for rectangle packing. Computers & Operations Research, 34(11), 3270–3280.

IBARAKI, T.; NAKAMURA, K. (2006). Packing problems with soft rectangles. In International Workshop on Hybrid Metaheuristics, pages 13–27. Springer.

JENIFER, J.; ANAND, S.; LEVINGSTAN, Y. (2016). Simulated annealing algorithm for modern VLSI floorplanning problem. ICTACT Journal on microelectronics, 2.

JI, P.; HE, K.; JIN, Y.; LAN, H.; LI, C. (2017). An iterative merging algorithm for soft rectangle packing and its extension for application of fixed-outline floorplanning of soft modules. Computers & Operations Research, 86, 110–123.

LASKAR, N.; SEN, R.; PAUL, P.; BAISHNAB, K. (2015). A survey on VLSI floorplanning: Its representation and modern approaches of optimization. IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems.

MARTELLO, S.; MONACI, M. (2015). Models and algorithms for packing rectangles into the smallest square. Computers & Operations Research, 63, 161–171.

MURATA, H.; KUH, E. S. (1998). Sequence-pair based placement method for hard/soft/pre-placed modules. In Proceedings of the 1998 international symposium on Physical design, pages 167–172. ACM.

SINGH, R.; BAGHEL, A.; AGARWAL, A. (2016). A review on VLSI floorplanning optimization using metaheuristic algorithms. International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), pages 4198–4202.

WANG, Y.; CHEN, L. (2015). Two-dimensional residual-space-maximized packing. Expert Systems with Applications, 42(7), 3297–3305.

WU, L.; ZHANG, L.; XIAO, W.-S.; LIU, Q.; MU, C.; YANG, Y. (2016). A novel heuristic algorithm for two-dimensional rectangle packing area minimization problem with central rectangle. Computers & Industrial Engineering, 102, 208–218.